Congratulations! You passed!

Grade received 100%
To pass 80% or higher

Go to next item

Graded Quiz: Mastering SQL Joins

Latest Submission Grade 100%

1. Which of this type of join returns an output of only matching records in the two tables? ***(Select all that apply)*** **1 / 1 point**

- [x] JOIN

  Correct
  Correct! This type of join returns those records which have matching values in both tables.

- [ ] LEFT JOIN
- [ ] FULL JOIN
- [x] INNER JOIN

  Correct
  Correct! This type of join returns those records which have matching values in both tables.

2. What is the golden rule for performing SQL joins? ***(Select all that apply)*** **1 / 1 point**

- [ ] Find a related row or record in both tables
- [x] Find a related column or field in both tables

  Correct
  Correct! One important thing to note is, we can only join these two tables on their related or linking column or field,

- [ ] Find the unique and primary keys in both tables
- [x] Find the linking field or column in both tables

  Correct
  Correct! One important thing to note is, we can only join these two tables on their related or linking column or field,

3. Consider the tables with their respective fields below: **1 / 1 point**

employees: emp_id, dept_id, manager_id, last_name

departments: dept_id, manager_id, dept_name, location_id

You want to create a report displaying employees last names, department names and locations, Which query should you use to create an INNER JOIN?

- ○

```
1   SELECT employees.last_name,departments.dept_name,departments.location_id
2   FROM employees e
3   INNER JOIN departments D
```

```
SELECT e.last_name, d.dept_name, d.location_id
FROM employees e
JOIN departments d
ON e.dept_id = d.dept_id;
```

```
SELECT last_name, dept_name, location_id
FROM employees, departments;
```

```
SELECT e.last_name, d.dept_name, d.location_id
FROM employees e
JOIN departments d;
```

Correct
Correct! This syntax is correct. It specifies correctly the tables and respective fields. In addition, the JOIN keyword and the ON predicate was specified correctly.

4. How many records are in the dept_manager_dup table? ***(Please enter a numeric value/answer)*** **1 / 1 point**

26

Correct
Correct! From the project, the dept_manager_dup table had 26 records.

5. The ON predicate in joins is written like a _____ **1 / 1 point**

- DEFINE clause
- SELECT clause
- WHERE clause
- FROM clause

Correct
Correct! The ON predicate works exactly like the WHERE clause. It returns in the output where the two fields match. In fact, WHERE used to be the old way of writing join statements. However, ON is used due to query optimization.

6. Which of these is true of LEFT JOIN? ***(Select all that apply)*** **1 / 1 point**

- [x] All records from the left table that have no matching record from the right table.

Correct
Correct! In addition to LEFT JOIN retrieving all matching records in both tables, it returns all values from the left table that match no values from the right table

- [x] It returns all matching records in both tables

Correct
Correct! The LEFT JOIN statement returns all matching records in both tables (that is, it works like INNER JOIN)

- [ ] All records from the right table

7. Consider two tables: **dept_manager and departments** **1 / 1 point**

Write a query that returns how many managers belong to different departments. ***(Select all that apply)***

- [x]
```
SELECT d.dept_name, COUNT(m.emp_no)
FROM dept_manager m
JOIN departments d
ON m.dept_no = d.dept_no
GROUP BY d.dept_name;
```

Correct

Correct! This query is syntactically correct. Since we need how many managers, the COUNT function is used. Additionally, the tables in question were correctly specified. Lastly, the GROUP BY clause was used because of the aggregate function used.

- [x]
```
SELECT d.dept_name, COUNT(m.emp_no)
FROM dept_manager m
JOIN departments d
ON m.dept_no = d.dept_no
GROUP BY dept_name;
```

Correct

Correct! This query is syntactically correct. Since we need how many managers, the COUNT function is used. Additionally, the tables in question were correctly specified. Lastly, the GROUP BY clause was used because of the aggregate function used.
Whether we GROUP BY d.dept_name or just dept_name, they mean the same thing.

- [ ]
```
SELECT d.dept_name, COUNT(m.emp_no)
FROM dept_manager m
JOIN department d
ON m.dept_no = d.dept_no
GROUP BY d.dept_name;
```

- [ ]
```
SELECT d.dept_name, COUNT(m.emp_no)
FROM dept_manager m
JOIN departments d
ON m.dept_no = d.dept_no;
```

8. **1 / 1 point**

EMPLOYEES

LAST_NAME	DEPARTMENT_ID	SALARY
Getz	10	3000
Davis	20	1500
King	20	2200
Davis	30	5000
Kochhar		5000

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Sales
20	Marketing
30	Accounts
40	Administration

Retrieve all employees last names, departments and salary, whether or not they have matching departments in the departments table.
Which query would you use?

- (•)
```
SELECT e.last_name, d.department_name, e.salary
FROM employees e
LEFT OUTER JOIN departments d
ON e.department_id = d.department_id;
```

```
SELECT e.last_name, d.department_name, e.salary
FROM employees e
RIGHT OUTER JOIN departments d
ON e.department_id = d.department_id;
```

```
SELECT e.last_name, d.department_name, e.salary
FROM employees e
FULL OUTER JOIN departments d
ON e.department_id = d.department_id;
```

Correct

Correct! This clearly shows a LEFT JOIN statement, since we are interested in joining the employees (the left table) to the departments (right table) whether they match or not with the departments table (right table)

9. Evaluate this SQL statement: **1 / 1 point**

```
SELECT e.employee_id, e.department_id, d.department_name, e.salary
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

Which SQL statement is equivalent to the above SQL statement?

```
SELECT e.employee_id, e.department_id, d.department_name, e.salary
FROM employees e
JOIN departments d
ON e.department_id = d.department_id;
```

```
SELECT employee_id, department_id, department_name, salary
FROM employees
JOIN departments
USING e.department_id, d.department_id;
```

```
SELECT employee_id, department_id, department_name, salary
FROM employees
WHERE department_id IN (SELECT department_id
FROM departments);
```

```
SELECT e.employee_id, e.department_id, d.department_name, e.salary
FROM employees
NATURAL JOIN departments;
```

Correct

Correct! Using JOIN and ON gives the same result as the query in the result. Using WHERE is called the **old join syntax.** Using JOIN and ON is referred to as the **new join syntax.**

Although using **WHERE or JOIN,**the retrieved output is identical. However, using WHERE is more time-consuming. Therefore, the WHERE syntax is perceived as **morally old** and is rarely employed by professionals. The JOIN syntax allows you to modify the connection between tables easily.

10. Which of the following is true about FULL OUTER JOIN created on two tables Table 1 and Table 2? **1 / 1 point**

- Retrieves both matched and unmatched rows of Table 1 and Table 2
- Retrieves the unmatched rows of Table 2

- Retrieves only matched rows of Table 1 and Table 2
- Retrieves all the unmatched rows of Table 1

Correct
Correct! In SQL the **FULL OUTER JOIN** combines the results of both left and right **outer joins** and returns all (matched or unmatched) rows from the tables on both sides of the **join** clause.

11. What is the proper way to write a DATE data type in SQL? **1 / 1 point**

- YYYY/MM/DD
- YYYY-MM-DD
- DD/MM/YYYY
- DD-MM-YYYY

Correct
Correct! The proper format of a DATA data type is YYYY-MM-DD im that order

12. How many aggregate functions are in SQL? **1 / 1 point**

- 4
- 3
- 6
- 5

Correct
Correct! There are 5 aggregate fucntions. COUNT is one of the aggregate functions in SQL. Other aggregate functions are SUM (to get the sum of a numeric column, e.g salary), AVG (to get the average of a numeric column, e.g salary), MIN (to retrieve the minimum value of a column), MAX (to retrieve the maximum value of a column). Due to how **aggregate functions** work, they are also called **summarizing functions.**

13. A record that has the same data value for every field of a table in SQL, is said to be a _____ record. **1 / 1 point**

- Distinct
- Similar
- Duplicate
- All of the above

Correct
Correct! In SQL, duplicate records also known as duplicate rows , are identical rows in an SQL table. A record that has the same data value for every field of the table. Duplicate records gives inaccurate or stale data, which leads to bad reporting, skewed metrics, and poor sender reputation.